ActionScripts in FlashMX

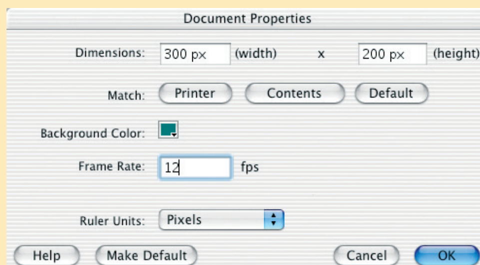
Bewegende beestjes

Flash-animaties brengen leven in de webbrouwerij. Beginnende Flash'ers durven meestal nog geen ActionScripts gebruiken. Toch vormen deze scripts heel vaak de motor van een Flash-animatie. Eenmaal je het principe onder de knie hebt, zal je dan ook aan ActionScripts de voorkeur geven om de snelheid en de richting van een beweging te bepalen.

Om zelf met deze workshop aan de slag te gaan, kan je een trialversie van FlashMX downloaden op [**www.macromedia.com/software/flash/download**]. Om het principe van beweging door ActionScripts te begrijpen zullen we een eenvoudige figuur een rechtlijnige beweging laten uitvoeren. In FlashMX-taal noemen we dat een beweging langs de X-as.

Stap 1
Aan de slag

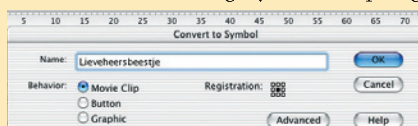
Je maakt een nieuw Flash-filmpje van bijvoorbeeld 300 bij 200 pixels met een groene achtergrond. Vervolgens importeer je het grafisch element. In dit voorbeeld hebben we uit de bijgeleverde voorbeelden (samples) het lieveheersbeestje gekozen. Dit heeft voor deze oefening geen belang, kies gewoon een leuke figuur.



Geef de afmetingen in van je nieuw filmpje.

Stap 2
Symbool maken

Deze figuur converteren we nu naar een symbool. Waarom? Heel eenvoudig: symbolen springen in je Flash-animatie erg zuinig met geheugen om. Om de verandering te voltooien gebruik je de functietoets **F8** of klik je op INSERT en vervolgens op CONVERT TO SYMBOL.



Gebruik liever symbolen dan afbeeldingen.

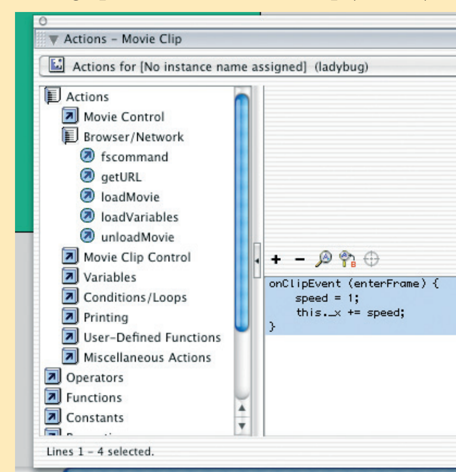
Stap 3
Tijd voor actie

Het geheel komt nog steeds zeer statisch over, dus het is de hoogste tijd om er wat meer beweging in te brengen. Met de rechtermuisknop klik je op het lieveheersbeestje en selecteer je ACTIONS. Je kan het Actions-venster ook oproepen door op WINDOW en vervolgens ACTIONS te klikken of de functietoets **F9** in te drukken. Terwijl het lieveheersbeestje geselecteerd blijft, geef je in het Actions-venster de volgende lijntjes code in:

```
onClipEvent(enterFrame) {
    speed = 1;
    this._x += speed;
}
```

En klaar is Kees. Bekijk de voorvertoning (CONTROL – TEST MOVIE) van de animatie en je zal zien dat het figuurtje traag, maar zonder schokken van links naar rechts beweegt.

Wat gebeurt er precies? Het eerste lijntje 'onClipEvent(enterFrame)' is niets meer dan een commando dat ervoor zorgt dat de code wordt uitgevoerd in een movieclip die slechts uit één frame bestaat. De code 'enterframe' kan je in mensentaal best vertalen als: 'voer onderstaande code uit.' In het tweede lijntje geef je aan dat de snelheid één moet zijn. Wanneer je daar 2, 3 of hoger invult zal de snelheid worden opgevoerd. Logisch, niet? De huidige positie van de movieclip op de X-as wordt dus telkens met één waarde verhoogd. Dit proces herhaalt zich een aantal keer per seconde en op die manier wordt beweging gesimuleerd. De snelheid waarmee de beweging wordt uitgevoerd hangt bovendien af van de framerate (hoe snel de frames per seconde worden afgespeeld). In de eerste schermafbeelding merk je dat we voor de normale norm gekozen hebben: 12 frames per seconde. Het derde lijntje maakt de optelling van de huidige positie van de movieclip (this._x), vermeerderd met de snelheidswaarde die je daarboven heb ingegeven. Je gebruikt de operator '+' gevolgd door '= speed' en zodoende weet je dat de waarde op de X-as continu zal verhogen. Wie opgelet heeft in de wiskundeles weet dat een hogere waarde op de X-as gewoon een beweging van links naar rechts betekent.



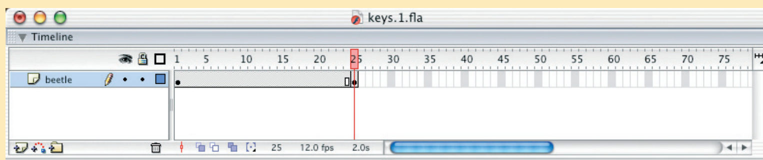
ActionScripts brengen leven in de brouwerij.

Stap 4
Experimenteren is fun!

Nu je dit alles weet, kan je een paar experimentjes uitvoeren met je lieveheersbeestje. Verhoog bijvoorbeeld de waarde van 'speed' met 2 in plaats van 1. De animatie verloopt dubbel zo snel. Laat het beestje de ziel uit zijn lijf rennen door het aan snelheid 3 of 4 te laten lopen. Een ander experiment: plaats het lieveheersbeestje aan de rechterkant en noteer als waarde voor de snelheid -2 in plaats van 2. Inderdaad: het beestje kruipt nu van rechts naar links. De waarde op de X-as vermindert alsmaar. Nog eentje om het af te leren: wanneer we in deze code de x in y veranderen, zal het beestje zich niet horizontaal maar verticaal bewegen, op de Y-as. In Flash neemt de Y-waarde af naarmate je hoger beweegt. Wanneer het object lager gaat, zal de Y-waarde toenemen.

Stap 5
Een beweging van 25 frames

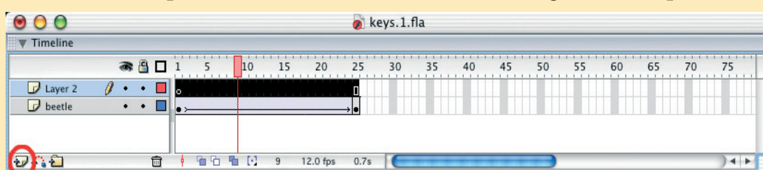
Je kan het lieveheersbeestje ook een niet-rechtlijnig pad laten volgen. In dat geval zullen we toch een beroep moeten doen op een zogenaamde 'Motion Tween' in plaats van een ActionScript. Plaats een nieuw keyframe op vakje 25: INSERT – KEYFRAME.



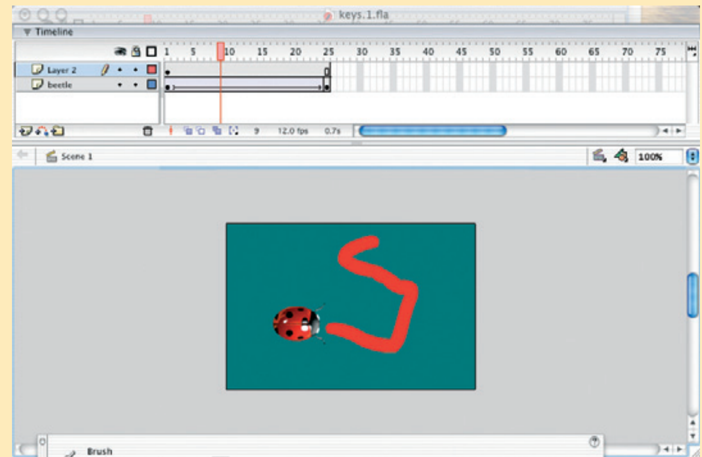
Keyframes sturen je beweging.

Stap 6
Bewegingsovergang

Vervolgens klik je in de tijdlijn ergens tussen vakje één en vijfentwintig. En in het Properties-venster geef je aan dat hier een MOTION TWEEN moet komen. Je kruist bovendien de optie ORIENT TO PATH aan, hierdoor zal het diertje zich meedraaien naar de richting van het pad dat je zodadelijk gaat tekenen.
Vervolgens voeg je een nieuwe laag toe door op het laagicoontje met het plusteken te klikken. Met het borstelgereedschap schilder je het pad ruwweg op deze nieuwe laag. Wees gerust, deze borstelstreek is in de uiteindelijke presentatie niet zichtbaar.



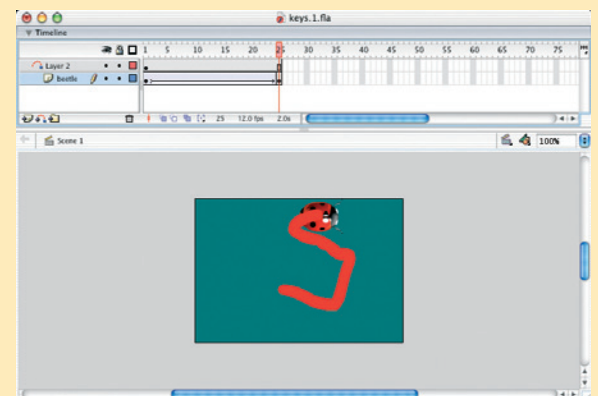
We bereiden de gidslaag voor.



Dit is de weg die het beestje gaat afleggen.

Stap 7
Volg de gids

Dan verander je de eigenschappen van deze laag via MODIFY – LAYER, en daar bepaal je dat de laag waarop je het pad geschilderd heb GUIDE wordt. Je klikt op de laag met het beestje, opnieuw selecteer je MODIFY – LAYER en daar duid je aan dat deze laag GUIDED wordt. Je klikt op de tijdlijn op het eerste frame en daarna sleep je het lieveheersbeestje met het cirkeltje op het begin van de borstelstreek. Daarna klik je op frame 25 en vervolgens sleep je het diertje op het einde van de borstelstreek. In de preview, die je oproept door te klikken op CONTROL – TEST MOVIE, merk je dat het beestje zich nu braaf in alle bochten wringt, zonder dat het pad zelf zichtbaar is.



Het lieveheersbeestje blijft op het 'rechte' pad.



Nog even het filmpje testen en klaar is kees.

— Dirk Schoofs —